Example 4.18

Example 4.18

1. According to the Chomsky hierarchy, CFG is type ———- grammar.
2. The grammar where production rules are in the format of $\{Astringconsistsofatleastonenon - terminal\} \rightarrow$ $\{Astringofterminalsand/ornon - terminals\}$ is ———- grammar in particular.
3. The grammar $S \rightarrow aSb/bSb/C$ produces the string ———- .
4. Finding a derivation for a string from a given grammar is called ———-.
5. The tree representation of deriving a context-free language from a given context grammar is called ———- .
6. A parse tree construction is only possible for ———- grammar.
7. The root of the parse tree of a given context-free language is represented by the ———- of the corresponding CFG.

8. A CFG G is said to be ———- if there exists some $w \in L(G)$ that has at least two distinct parse trees.

9. A CFL L is said to be ———- if all its grammars are ambiguous.

10. The CFG where a non-terminal $'A'$ as a leftmost symbol appears alternatively at the time of derivation either immediately or through some other non-terminal definitions is called ———- grammar.

11. To avoid the problem of backtracking, we need to perform ———- .

12. In a CFG, the symbols which do not produce any terminal string is called ———- .

13. In a CFG, the symbols which cannot be reached at any time starting from the start symbol are called ———- .

14. In a CFG, non-generating symbols and non-reachable symbols are both called ———- symbol.

15. In a CFG, the production in the form non-terminal $\rightarrow$ single non-terminal is called .

16. In a CFG, a production in the form $NT \rightarrow \in$ is called ——— .

17. Normalizing a CFG should not hamper the ——— power of the grammar.

18. A CFG where all the productions of the grammar are in the form

Non-terminal $\rightarrow$ string of exactly two non-terminals

Non-terminal $\rightarrow$ single terminal

is called ——— normal form.

19. A CFG where all the productions of the grammar are in the form

Non-terminal $\rightarrow$ (single terminal)(string of non-terminals)

Non-terminal $\rightarrow$ single terminal

is called ——— normal form.

20. Context-free languages are not closed under ——— and ——— .

21. ——— is used to prove that certain sets are not context free.

22. $a^n b^n c^n$, where $n \geq 1$, is not ———- language but ———- language.
23. If the length of the longest path of the directed graph generated from a CFG is n, then the longest string generated by the grammar is ———- .
24. A language L generated from a given CFG is finite if there are no ———- in the directed graph generated from the production rules of the given CFG.

Fill in the Blanks
Fill in the Blanks
Exercise
picture
Exercise
picture
Exercise
Exercise
Exercise
Exercise

22. $a^n b^n c^n$, where $n \geq 1$, is not ——— language but ——— language.
23. If the length of the longest path of the directed graph generated from a CFG is n, then the longest string generated by the grammar is ——— .
24. A language L generated from a given CFG is finite if there are no ——— in the directed graph generated from the production rules of the given CFG.

Answers:

1. Two
2. context-free
3. $WCW^R \mid W \in (a, b)^*$
4. parsing
5. parse tree
6. context-free
7. start symbol
8. ambiguous
9. inherently ambiguous
10. left recursive
11. left factoring
12. non-generating symbols
13. non-reachable symbols
14. useless
15. unit production
16. null production
17. language generating
18. Chomsky
19. Greibach
20. intersection, complementation
21. Pumping lemma for CFL
22. context free, context sensitive
23. 2^n
24. cycles

Exercise

1. Construct a CFG for the following.
   a) $a^n a^m$, where $n > 0$ and $m = n + 1$
   b) $a^n ba^m$, where $m, n > 0$
   c) $a^n b^n c^m$, where $n > 0$ and $m = n + 1$
   d) $L = (011 + 1) * (01)*$
   e) $L = \{Setofallintegers\}$
2. a) Construct the string 0110001 from the grammar

$$S \rightarrow AB$$
$$A \rightarrow 0A/1B/0$$
$$B \rightarrow 1A/0B/1$$

By using

i) Leftmost derivation

ii) Rightmost derivation

b) Construct the string *baaabbba* from the grammar

$$S \to AaB/AbB$$
$$A \to Sa/b$$
$$B \to Sb/a$$

By using

i) Leftmost derivation

ii) Rightmost derivation

b) Construct the string *baaabbba* from the grammar

$$S \to AaB/AbB$$
$$A \to Sa/b$$
$$B \to Sb/a$$

By using

i) Leftmost derivation

ii) Rightmost derivation

3. a) Find the parse tree for generating the string *abaabaa* from the given grammar.

$$S \to aAS/a$$
$$A \to bS$$

b) Find the parse tree for generating the string aabbaa from the given grammar.

$$S \to aAS/a$$
$$A \to SbA/SS/ba$$

b) Find the parse tree for generating the string aabbaa from the given grammar.

$$S \rightarrow aAS/a$$
$$A \rightarrow SbA/SS/ba$$

4. Show that the following grammars are ambiguous.

a) $S \rightarrow abSb/aAb/a$

$A \rightarrow bS/aAAb$

b) $E \rightarrow E + E/E * E/id$

c) $S \rightarrow aB/bA$

$A \rightarrow aS/bAA/a$

$B \rightarrow bS/aBB/b$

5. Remove the useless productions from the following grammar

a) $S \rightarrow AB/a$

$A \rightarrow b$

b) $S \rightarrow AB/AC$

$A \rightarrow 0A1/1A0/0$

$B \rightarrow 11A/00B/AB$

$C \rightarrow 01C0/0D1$

$D \rightarrow 1D/0C$

5. Remove the useless productions from the following grammar

a) $S \to AB/a$

$A \to b$

b) $S \to AB/AC$

$A \to 0A1/1A0/0$

$B \to 11A/00B/AB$

$C \to 01C0/0D1$

$D \to 1D/0C$

6. Remove the unit production from the following grammar:

a) $S \to SaA$

$A \to aB/B/b$

$B \to bC/C/a$

$C \to ab$

b) $S \to Aa/B$

$B \to A/bb$

$A \to a/bc/B$

5. Remove the useless productions from the following grammar

a) $S \to AB/a$

$A \to b$

b) $S \to AB/AC$

$A \to 0A1/1A0/0$

$B \to 11A/00B/AB$

$C \to 01C0/0D1$

$D \to 1D/0C$

6. Remove the unit production from the following grammar:

a) $S \to SaA$

$A \to aB/B/b$

$B \to bC/C/a$

$C \to ab$

b) $S \to Aa/B$

$B \to A/bb$

$A \to a/bc/B$

7. Remove the null production from the following grammar

a) $S \to aAB$

$A \to Bb$

$B \to \in$

b) $S \to aA$

$A \to bB$

$B \to b$

$B \to \in$

8. Simplify the following CFG.

$$S \rightarrow AB/aB$$
$$A \rightarrow BC/B/a$$
$$B \rightarrow C$$
$$C \rightarrow b/\in$$

8. Simplify the following CFG.

$$S \rightarrow AB/aB$$
$$A \rightarrow BC/B/a$$
$$B \rightarrow C$$
$$C \rightarrow b/\in$$

9. i) Convert the following left linear grammar into right linear grammar.

$$S \rightarrow Sab/Aa$$
$$A \rightarrow Abb/bb$$

ii) Convert the following right linear grammar into left linear grammar.

$$S \rightarrow aaB/ab$$
$$B \rightarrow bB/bb$$

8. Simplify the following CFG.

$$S \to AB/aB$$
$$A \to BC/B/a$$
$$B \to C$$
$$C \to b/\in$$

9. i) Convert the following left linear grammar into right linear grammar.

$$S \to Sab/Aa$$
$$A \to Abb/bb$$

ii) Convert the following right linear grammar into left linear grammar.

$$S \to aaB/ab$$
$$B \to bB/bb$$

10. Convert the following linear grammar into regular grammar.

$$S \to 01B/0$$
$$B \to 1B/11$$

Fill in the Blanks
Fill in the Blanks
Exercise
picture
Exercise
picture
Exercise
Exercise
Exercise
Exercise

11. Convert the following grammar into CNF.

a) $S \rightarrow AB$

$A \rightarrow aA/a$

$B \rightarrow ab/bB/b$

b) $S \rightarrow aSa/SSa/a$

11. Convert the following grammar into CNF.

    a) $S \rightarrow AB$

    $A \rightarrow aA/a$

    $B \rightarrow ab/bB/b$

    b) $S \rightarrow aSa/SSa/a$

12. Convert the following grammar into GNF.

    a) $S \rightarrow Abb/a$

    $A \rightarrow aaA/B$

    $B \rightarrow bAb$

    b) $S \rightarrow aSb/bSa/a/b$

13. Construct a DFA equivalent to the regular grammar.

a) $S \to aS/bA/b$

$A \to aA/bS/a$

b) $S \to bA/aB$

$A \to bA/aS/a$

$B \to aB/bS/b$

14. Prove that $L = a^n b^n c^2 n$ is not context free by using the pumping lemma for CFL.

15. Verify whether the languages generated by the following grammar are finite or not.

a) $SßaA$

$AßBC$

$BßSC/b$

$CßB/a$

b) $SßAB$

$AßC/a$

$BßAC/b$

16. Remove the left recursion from the following grammar and then perform left factoring.

$$E \rightarrow E + T | T$$
$$T \rightarrow T * F | F$$
$$F \rightarrow G^F | G$$
$$G \rightarrow id | (E)$$

16. Remove the left recursion from the following grammar and then perform left factoring.

$$E \rightarrow E + T|T$$
$$T \rightarrow T * F|F$$
$$F \rightarrow G^F|G$$
$$G \rightarrow id|(E)$$

17. Generate the string $id + id * id$ from the grammar

$$E \rightarrow E + E$$
$$E \rightarrow E * E$$
$$E \rightarrow id$$

16. Remove the left recursion from the following grammar and then perform left factoring.

$$E \to E + T | T$$
$$T \to T * F | F$$
$$F \to G^F | G$$
$$G \to id | (E)$$

17. Generate the string $id + id * id$ from the grammar

$$E \to E + E$$
$$E \to E * E$$
$$E \to id$$

where the precedence of operator is given as follows.

	$+$	$*$
$+$	$>$	$<$
$*$	$>$	$<$

Are you getting any ambiguity in the grammar?